

## ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ ЗНАЧЕНИЙ ПОТЕНЦИАЛОВ КОРПУСНЫХ СТАЛЕЙ В МОРСКОЙ ВОДЕ

**В.В. Альчаков**

*ФГАОУ ВО «Севастопольский государственный университет»,*

*г. Севастополь, Россия*

*ORCID: 0000-0002-1392-1699*

**В.А. Крамарь**

*ФГАОУ ВО «Севастопольский государственный университет»,*

*г. Севастополь, Россия*

*ORCID: 0000-0002-0528-1978*

**А.В. Родькина**

*ФГАОУ ВО «Севастопольский государственный университет»,*

*г. Севастополь, Россия*

*ORCID: 0000-0002-4593-4259*

*Аннотация. В статье приведен анализ существующих моделей нейронных сетей. Описаны особенности построения нейронной сети с использованием нескольких параметров, влияющих на выходную величину. Выявлены преимущества использования нейронных сетей и вычислительных систем на их основе. Поставленная задача была решена с использованием языка программирования Python. Разработана вычислительная модель значений потенциала стали с оксидной пленкой и потенциала стали без оксидной пленки для различной солёности морской воды и различных видов корпусных сталей, применяемых для корпусов морских судов и подводных конструкций океанотехнических сооружений, проектируемых для Черноморского бассейна Севастопольского региона и эксплуатирующихся в этом регионе. Полученные результаты работы позволят повысить точность прогнозирования потенциалов для различных марок корпусных сталей.*

*Ключевые слова: нейронные сети, корпусные стали, морские суда, океанотехнические сооружения, потенциал, слои, веса, пороги.*

### Введение

Постоянное развитие математического и алгоритмического аппарата нейронных сетей (НС) открывает новые возможности для получения высокоточных моделей различных процессов и явлений, наблюдаемых в морской среде. Среди таких процессов можно выделить явления в металлических конструкциях и сооружениях морского назначения, для которых исследование корреляции параметров, влияющих на надежность конструкций, является актуальной задачей, решение которой может быть получено с помощью мощнейшего аппарата НС. При этом НС могут быть использованы для изучения и исследования как статистических, так и динамических свойств объектов, что дает возможность получения точных оценок неизвестных параметров на основании исторических данных об исследуемом процессе или явлении. В работе [1] получены результаты применения НС для прогнозирования величины потенциала стали на ювенильной поверхности, однако при расчетах не было учтено влияние катодной поляризации, что могло повлиять на точность полученных нейросетевых моделей и соответствующих оценок исследуемых параметров, а соответственно не позволяет обеспечить должную защиту от локальных форм коррозионно-механических разрушений корпусных сталей морских судов и океанотехнических сооружений, проектируемых для Черноморского бассейна Севастопольского региона и эксплуатирующихся в этом регионе.

В качестве эффективного инструмента для изучения процесса коррозии были использованы многие методы, основанные на алгоритме машинного обучения, такие как модель нейронной сети обратного распространения (BPNN), модель нейронной сети с радиальной базисной функцией (RBF), модель опорных векторов (SVM), модель самоорганизующихся карт (SOM) и модель теории Грея. Интеллектуальный алгоритм часто используется для прогнозирования плотности коррозии в стальных железобетонных конструкциях, потенциала поляризации, плотности рассеянного тока, потери массы [2], скорости коррозии, классификации опасности коррозии и т. д. НС широко используется в области прогнозирования коррозии. Основным преимуществом НС являются существенные возможности в установлении нелинейных связей между входными и выходными наборами данных о коррозии.

НС играют большую роль как средство для внедрения экспертных систем на основании того, что даже с одним скрытым слоем и соответствующим образом выбранным числом нейронов НС может приближаться к любой непрерывной функции.

Основой НС является искусственный нейрон, который был разработан, чтобы имитировать характеристику биологического нейрона. МакКаллок и Питс предложили двоичную пороговую единицу в качестве вычислительной модели для нейрона. Схематическая диаграмма нейрона показана на рис. 1 [3].

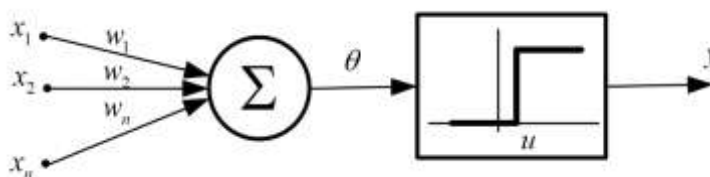


Рис. 1. Схематическая диаграмма нейрона

Принцип действия нейрона, изображенного на рис. 1 состоит в следующем. На вход нейрона поступают сигналы  $x_i$ ,  $i = 1, 2, \dots, n$  при этом каждый сигнал имеет свой вес  $w_i$ . Сумма взвешенных входных сигналов сравнивается с заданным пороговым значением  $u$ , если сумма превышает  $u$ , величина выходного сигнала принимается равной «1», в противном случае на выходе формируется «0». Математическая запись алгоритма работы нейрона имеет вид

$$y = \theta \left( \sum_{i=1}^n w_i x_i - u \right),$$

где  $\theta(\cdot)$  – пороговая функция (которая представляет функцию активации),  $w_i$  – вес синапса  $p$ , связанный с  $j$ -м входом [3].

Доказано, что при правильно выбранных весах синхронное расположение таких нейронов в принципе способно к универсальным вычислениям.

При решении различных задач могут использоваться функции активации, отличные от указанной пороговой функции, например, кусочно-линейная, сигмоидальная или гауссовская.

Пороговая функция определяется как

$$y = \theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Логистическая функция является стандартной сигмоидальной функцией и определяется как

$$y = \theta(x) = \frac{1}{1 + e^{-\beta x}}. \quad (1)$$

Сигмовидная функция является наиболее часто используемой функцией в НС. Это гладкая строго возрастающая непрерывная функция.

Гиперболический тангенс определяется как

$$y = \theta(x) = \tanh(x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}. \quad (2)$$

Кусочно-линейная функция определяется с помощью следующего соотношения

$$y = \theta(x) = \begin{cases} 0 & \text{if } x \leq -1 \\ \frac{x+1}{2} & \text{if } -1 \leq x \leq 1 \\ 1 & \text{if } x \geq 1 \end{cases}$$

Гауссова функция определяется выражением

$$y = \theta(x) = \exp\left(\frac{-x^2}{2\sigma^2}\right),$$

где  $\sigma$  – дисперсия, параметр, значение которого управляет свойствами гладкости интерполирующей функции.

НС можно рассматривать как взвешенные ориентированные графы, в которых узлы являются искусственными нейронами и направленные ребра с весами являются соединениями от выходов нейронов до входов нейронов. Архитектура соединения НС могут быть сгруппированы в две основные категории [4]. Первая категория – это сети с прямой связью, в которой на графике нет петель, а вторая – сети с обратной связью (или рекуррентные сети), в которых существуют петли из-за соединений с обратной связью.

Наиболее распространенное семейство сетей с прямой связью – это многоуровневая нейронная сеть, в которой нейроны организованы в слои со связями строго в одном направлении от одного уровня к другому.

Разные возможности подключения приводят к разному поведению нейронных сетей. Нейронные сети с прямой связью являются статическими сетями, т. е. при вводе они генерируют только один набор выходных значений, а не последовательность значений. Сети с прямой связью не зависят от предыдущего состояния сети. Рекуррентные сети являются динамическими системами. При представлении нового шаблона ввода вычисляются выходы нейронов. Из-за обратной связи входы в каждый нейрон модифицируются, что приводит сеть в новое состояние.

Наиболее популярными искусственными нейронными сетями являются многослойные перцептроны. Также рассматриваются НС с радиальной базисной функцией (RBFNN). Эти сети доказали свою универсальность в качестве аппроксиматора, что позволяет использовать их для аппроксимации непрерывных многомерных функций с заданной степенью точности (при наличии необходимого количества нейронов в скрытом слое НС) [5].

Рассмотрим два типа нейронных сетей с прямыми связями [6, 7], которые получили наиболее широкое распространение, также описывается преимущество использования нейронных сетей с практической точки зрения.

### Нейронные сети прямого распространения

*Однослойный перцептрон.* Простейшая из существующих нейронных сетей представляет собой так называемый перцептрон. Перцептрон имеет много входов полностью соединенных с выходным слоем посредством нескольких нейронов МакКаллока-Питса [8] (рис. 2).

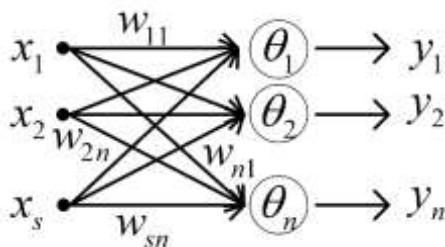


Рис. 2. Перцептрон с s входами и n выходами

Каждый вход  $x_i$  умножается на весовой коэффициент  $w_{ij}$  перед тем как попасть в обрабатывающий элемент  $\theta_j$  и на выходной слой  $y_j$

$$y_j = \theta(\text{net}_j) = \theta_j \left( \sum_{i=1}^s w_{ij} x_i + b_j \right), \quad j = 1, 2, \dots, n \quad (3)$$

В соотношении (3)  $\theta_j$  представляет собой пороговую функцию, а параметр  $b_j$  – соответствующая величина смещения. Количество выходов как правило определяется числом классов, на которые разбиты входные данные. Эти элементы обеспечивают масштабирование всего пространства входных сигналов и их влияние на выход системы.

Цель перцептрона проиллюстрирована на рис. 3. На рисунке показаны два класса входных данных (в простейшем случае можно использовать бинарную классификацию, тогда нулем можно обозначить один класс, а единицей – другой). Задача использования перцептрона сводится к поиску уравнения гиперплоскости, которая позволит однозначно отделить экземпляры класса 0 от экземпляров класса 1. В двухмерном пространстве такая гиперплоскость представляет собой линию, в трехмерном – плоскость, в s-мерном пространстве – поверхность размерности s-1 [7].

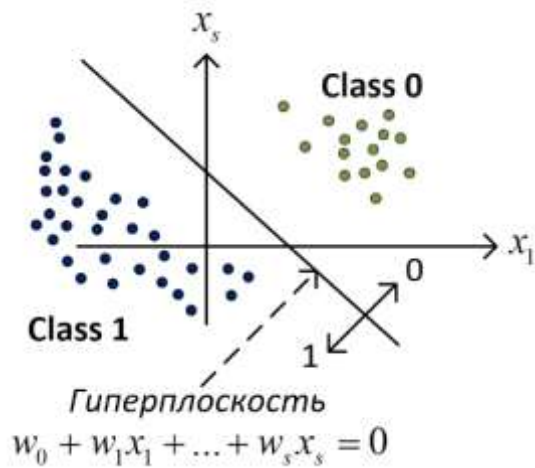


Рис. 3. Представление задачи классификации

Для определения набора весов  $w_0, w_1, \dots, w_s$  для каждого выхода перцептрона (каждый из которых будет соответствовать одному из классов на которые разделены входные данные). Вектор весов  $w$  перцептрона определяет гиперплоскость в  $s$ -мерном пространстве. Если одна из точек  $x_0, x_1, \dots, x_s$  лежит в плоскости, элементы которой классифицируются как класс 0, то выход перцептрона также равен 0. Если точка лежит в плоскости, содержащей элементы класса 1, то выход перцептрона также равен 1. В приведенном на рис. 3. случае перцептрон будет работать правильно благодаря тому, что гиперплоскость сориентирована правильно с точки зрения выявления двух линейно-разделяемых классов.

**Многослойный перцептрон.** Многослойный перцептрон является расширением однослойного перцептрона, внутри которого появляются новые слои, которые не подключены непосредственно к выходам системы. На рис. 4. представлен многослойный перцептрон с одним скрытым слоем. Таким образом, всего можно выделить три различных слоя: входной (с  $s$  входами), скрытый (содержит  $n$  слоев) и выходной ( $l$  слоев) [7, 9].

Каждый элемент входного уровня может быть соединен с элементом скрытого уровня с помощью связи, имеющей вес  $w_{ij}$ , описываемый вещественным числом. В большинстве случаев каждый входной элемент имеет связь с элементом скрытого уровня. Предположим, что входной сигнал  $x_i$  соединен с  $j$ -м элементом скрытого уровня, при этом вес этой связи равен  $w_{ij}$ . Это означает что на вход  $j$ -го элемента скрытого уровня поступит не сигнал  $x_i$ , а сигнал, величина которого определяется выражением  $x_i w_{ij}$ . Тогда, общий сигнал, который поступает на вход  $j$ -го элемента скрытого уровня может быть описан с помощью соотношения

$$\sum_{i=1}^s x_i w_{ij}.$$

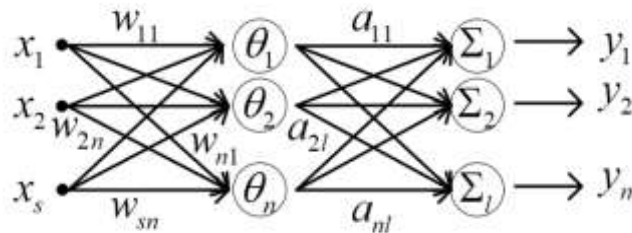


Рис. 4. Структура многослойного перцептрона

Далее,  $j$ -й элемент скрытого уровня (обозначим его  $\theta_j$ ) обрабатывает входной суммарный сигнал по определенному правилу. Обычно это правило описывается непрерывной сигмоидной функцией и называется функцией активации. Как правило, функция активации может быть записана как логистическая функция (1) или функция гиперболического тангенса (2).

Сигнал, полученный на выходе  $p$ -го элемента скрытого слоя, передается на  $l$ -й элемент выходного слоя с некоторым весом  $a_{pl}$ , который также описывается вещественным числом. Тогда, каждый выходной элемент нейронной сети представляет собой линейную комбинацию сигмоидных функций [7]. В этом общем случае для элемента на выходе внешнего слоя сети можно записать следующее выражение

$$y_k = \sum_{j=1}^n a_{kj} \theta_j \left( \sum_{i=1}^s w_{ij} x_i + b_j \right), \quad k = 1, 2, \dots, l \quad (4)$$

Соотношение (4) может быть обобщено на более общий случай, когда количество скрытых слоев нейронной сети больше 1.

Для повышения эффективности нейронной сети выполняется нормирование значений обучающей выборки:

$$\xi' = \frac{\xi - \xi_{min}}{\xi_{max} - \xi_{min}} \quad (5)$$

где  $\xi$  – необработанные данные образца,

$\xi_{min}$  и  $\xi_{max}$  – максимальные и минимальные исходные данные образца соответственно

$\xi'$  – нормированные данные между 0 и 1.

### Построение НС с использованием нескольких параметров, влияющих на выходную величину

Наборы данных, используемые для обучения нейронных сетей, достаточно часто используют сведения лишь об одном параметре, влияющем на величину неизвестного выходного параметра. Для повышения точности расчетов иногда, при наличии технических возможностей предлагается метод повышения качества прогноза, с помощью введения в обучающую выборку дополнительных параметров, влияющих на прогнозируемый параметр.

При решении задач прогнозирования временных рядов, как правило, используются структуры НС, представляющие собой многослойные, в частности, трехслойные НС. Обобщенная структура такой сети представлена на рис. 5.

В таком случае в качестве входных сигналов, поступающих на вход НС, являются значения параметров, которые определяют наиболее значимые факторы, процессы и явления, влияющие на формирование выходного сигнала, называемым прогнозом или оценкой неизвестного выходного параметра. Количество входных параметров может изменяться в диапазоне от 1 до  $N$  (где  $N$  – натуральное число). Количество нейронов скрытого слоя обозначается через  $M$  и подлежит определению.

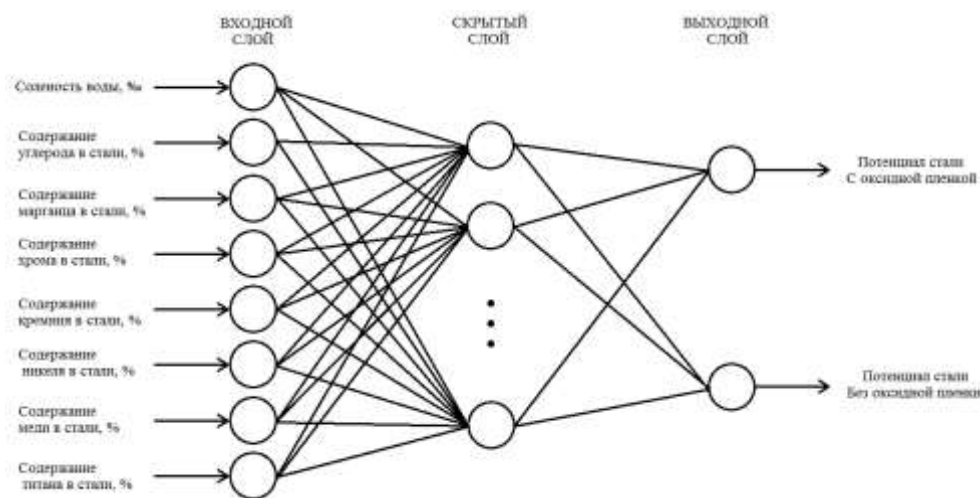


Рис. 5. Структура нейронной сети для прогнозирования

С точки зрения математического описания, постановка задачи может быть представлена в виде.

Пусть задана некоторая выборка  $\{x_i, y_i\}_{i=1}^n$ , где  $x_i$  – вектор значений входных параметров, влияющих на выходной параметр  $y_i$ . Величина параметра  $n$  определяет размер такой выборки с точки зрения доступных исторических значений наблюдаемого процесса или явления. Ставится задача построения НС, которая будет использоваться для получения прогноза выходного параметра в виде  $\hat{y} = f(x)$ . В случае, если число входных или выходных параметров больше 1 (что соответствует многофакторному или многомерному случаю) соответствующий прогноз может быть записан в матричном виде  $\hat{Y} = f(X)$ .

Следует учитывать, что имеющиеся в наличии исторические данные будут использоваться не только для обучения, но также и для верификации модели. При этом не рекомендуется использовать исторические данные обучающей выборки для тестирования и наоборот. В противном случае может произойти так называемое переобучение модели, что значительно ухудшит точность получаемых оценок для новых входных данных.

Исходные исторические данные не могут быть использованы «как есть», перед обучением НС прибегают к различным методам предварительной обработки данных, в частности, к масштабированию и нормированию исходной последовательности. В частности, соотношение (5) может быть использовано для нормирования каждого из входных и выходных параметров в диапазоне [0...1]. Такой подход является рекомендуемым и его применение позволяет повысить как качество, так и скорость обучения НС, что очень важно, например, для систем реального времени, в которых используется переобучение моделей с учетом новых исторических данных.

После того, как закончена предварительная обработка данных, переходят к расчётам количественных характеристик НС. К таким характеристикам относятся число входов НС, число выходов НС, а также длина исторической выборки, которая будет использована для обучения НС. Предположим, что проектируемая сеть будет иметь один скрытый слой и один выходной параметр, тогда, обозначив через  $N_x$  число входных параметров сети, влияющих на выходной параметр, а через  $N_y=1$  число выходов НС можно записать выражение для определения необходимого числа синоптических весов  $N_w$ , которое должно удовлетворять неравенству [10]

$$\frac{N_y Q}{1 + \log_2(Q)} \leq N_w \leq N_y \left( \frac{Q}{N_x} + 1 \right) (N_x + N_y + 1) + N_y.$$

Далее, необходимое число нейронов скрытого слоя может быть получено по формуле

$$N = \frac{N_w}{N_x + N_y}.$$

Исходя из полученных значений параметров  $N_w$  и  $N$  можно воспользоваться неравенством

$$\frac{Q}{10} - N_x - N_y \leq N \leq \frac{Q}{2} - N_x - N_y,$$

где  $Q$  – размер выборки, используемой для обучения НС, который равняется 75–80% от размера общей исторической выборки. Остальные 25–20% данных используются для верификации работы построенной НС.

### **Преимущества использования нейронных сетей и вычислительных систем на их основе**

Перечислим основные преимущества использования нейронных сетей по сравнению со стандартными вычислительными схемами.

**Адаптивные свойства.** Нейронные сети способны изменять свои параметры в соответствии с меняющимися условиями функционирования, т.е. обладают способностью решать новые задачи на основе данных, которые были представлены для обучения или начальной подготовки сети.

**Самоорганизация.** Искусственная нейронная сеть может самостоятельно создать или изменить свою собственную организацию на основании той информации, которая поступает в момент обучения или переобучения сети.

**Функционирование в режиме реального времени.** Вычисления искусственных нейронных сетей могут выполняться параллельно, при этом вычисления могут быть выполнены с помощью специализированных аппаратных вычислительных устройств в реальном масштабе времени. Это позволяет использовать вычислительные процедуры, в основе которых лежат нейронные сети, на бортовых устройствах различных устройств, требующих получения сигналов управления в реальном масштабе времени.

**Высокая отказоустойчивость.** Благодаря тому, что при обучении нейронных сетей зачастую используется избыточная информация, отказоустойчивость систем на их основе возрастает. Также частичное разрушение нейронной сети приводит к снижению производительности, однако может быть компенсировано за счет неповрежденных участков.

**Параллелизм.** Параллелизм является фундаментальным в архитектуре нейронных сетей, когда нейроны рассматриваются как наборы элементарных единиц. Работая одновременно, можно получить большую скорость вычисления.

### **Практическая реализация построения вычислительной модели значения потенциалов корпусных сталей в морской воде**

В последнее время для исследования данных и построения нейросетевых вычислительных моделей ученые и инженеры всего мира прибегают к использованию высокоуровневого языка программирования Python. Возможности этого языка делают его незаменимым инструментом инженеров и исследователей, работающих с большими данными, в том числе и для построения нейронных сетей на основе этих данных [11]. Одним из важнейших преимуществ Python является наличие большого количества свободно

распространяемых пакетов для анализа и визуализации данных, которые значительно упрощают сам процесс исследования и обработки данных. При этом сами пакеты, как и базовое ядро Python, позволяют получать кроссплатформенные решения без привязки и необходимости адаптации к какой-либо конкретной операционной системе, платформе или фреймворку. Перечисленные особенности Python позволяют расширить область применения языка за счет таких пакетов как NumPy (пакет для работы с одномерными данными в виде массивов), Pandas (пакет для работы с неоднородными и поименованными данными), SciPy (пакет для решения общих научных вычислительных задач), Matplotlib (пакет для визуального представления наборов данных), Scikit-Learn (пакет для решения задач машинного обучения), IPython и Jupyter (пакет, позволяющий создать вычислительную интерактивную среду, для легкого обмена созданными скриптами).

Именно этот набор инструментов далее будет использован для апробации предложенных подходов и проверки сделанных теоретических утверждений.

Цель построения вычислительной модели: прогнозирование значений потенциалов различных видов корпусных сталей с оксидной пленкой и без оксидной пленки при различной солености морской воды, применяемых для корпусов морских судов и подводных конструкций океанотехнических сооружений.

Для формирования вычислительной модели значения потенциалов корпусных сталей в морской воде нужна пара векторов: входные и выходные параметры. Для достижения поставленной цели вектор входных параметров состоит из восьми элементов: солености морской воды, процентного содержания углерода и легирующих элементов (марганец, хром, кремний, никель, медь, титан) в стали. Вектор выходных параметров состоит из двух элементов: потенциала стали с оксидной пленкой, мВ; потенциала стали без оксидной пленки, мВ (таблица 1, 2 [12]).

Таблица 1

Выборка для обучения нейронной сети

Марка стали	Соленость воды, ‰	Содержание в стали, %							Потенциал стали -ф, мВ					
		углерод	марганец	хром	кремний	никель	медь	титан	с оксидной пленки	без оксидной пленки				
		<i>В</i>	<i>Х</i>	<i>О</i>	<i>Д</i>	<i>Н</i>	<i>О</i>	<i>Й</i>	<i>С</i>	<i>Л</i>	<i>О</i>	<i>Й</i>	<i>ВЫХОДНОЙ СЛОЙ</i>	
Cr3	18	0.18	0.53	0.3	0.23	0.3	0.3	0	494.00	410.17				
09Г2	18	0.12	1.6	0.3	0.27	0.3	0.3	0	520.33	435.17				
20Х13	18	0.21	0.8	13	0.8	0	0	0	171.33	469.00				
10ХСНД	18	0.12	0.65	0.75	0.95	0.65	0.5	0	529.67	440.33				
12Х18Н10Т	18	0.12	2	18	0.8	10	0	0.7	448.00	491.00				
Cr3	20	0.18	0.53	0.3	0.23	0.3	0.3	0	510.00	438.33				
09Г2	20	0.12	1.6	0.3	0.27	0.3	0.3	0	538.67	441.50				
20Х13	20	0.21	0.8	13	0.8	0	0	0	123.00	542.17				
10ХСНД	20	0.12	0.65	0.75	0.95	0.65	0.5	0	551.33	457.67				
09Г2	22	0.12	1.6	0.3	0.27	0.3	0.3	0	534.67	447.50				
20Х13	22	0.21	0.8	13	0.8	0	0	0	162.00	509.67				
10ХСНД	22	0.12	0.65	0.75	0.95	0.65	0.5	0	556.00	457.17				
12Х18Н10Т	22	0.12	2	18	0.8	10	0	0.7	436.67	492.33				
Cr3	24	0.18	0.53	0.3	0.23	0.3	0.3	0	562.33	437.33				
20Х13	24	0.21	0.8	13	0.8	0	0	0	138.00	534.00				
10ХСНД	24	0.12	0.65	0.75	0.95	0.65	0.5	0	517.00	439.17				
12Х18Н10Т	24	0.12	2	18	0.8	10	0	0.7	430.33	536.83				
Cr3	26	0.18	0.53	0.3	0.23	0.3	0.3	0	519.67	415.67				
09Г2	26	0.12	1.6	0.3	0.27	0.3	0.3	0	513.00	431.83				
10ХСНД	26	0.12	0.65	0.75	0.95	0.65	0.5	0	539.00	452.50				
12Х18Н10Т	26	0.12	2	18	0.8	10	0	0.7	440.33	525.50				
Cr3	28	0.18	0.53	0.3	0.23	0.3	0.3	0	507.00	424.33				
09Г2	28	0.12	1.6	0.3	0.27	0.3	0.3	0	525.33	434.17				
20Х13	28	0.21	0.8	13	0.8	0	0	0	164.67	466.83				
12Х18Н10Т	28	0.12	2	18	0.8	10	0	0.7	441.00	516.50				
Cr3	30	0.18	0.53	0.3	0.23	0.3	0.3	0	505.33	432.33				
09Г2	30	0.12	1.6	0.3	0.27	0.3	0.3	0	527.33	428.17				
20Х13	30	0.21	0.8	13	0.8	0	0	0	167.33	458.00				
10ХСНД	30	0.12	0.65	0.75	0.95	0.65	0.5	0	535.33	441.17				
12Х18Н10Т	30	0.12	2	18	0.8	10	0	0.7	437.33	504.00				

Таблица 2



### Выборка для проверки нейронной сети

Марка стали	Соленость воды, %	Содержание в стали, %							Потенциал стали -ф, мВ					
		углерод	марганец	хром	кремний	никель	медь	титан	с оксидной пленки	без оксидной пленки				
		В	Х	О	Д	Н	О	Й	С	Л	О	Й	ВЫХОДНОЙ СЛОЙ	
12X18H10T	20	0.12	2	18	0.8	10	0	0.7	356.67	347.17				
Ст3	22	0.18	0.53	0.3	0.23	0.3	0.3	0	538.00	434.33				
09Г2	24	0.12	1.6	0.3	0.27	0.3	0.3	0	502.00	428.33				
20Х13	26	0.21	0.8	13	0.8	0	0	0	167.00	459.00				
10ХСНД	28	0.12	0.65	0.75	0.95	0.65	0.5	0	533.67	446.67				

Помимо построения моделей, немаловажную роль играет предварительная подготовка и анализ данных. Исходные данные могут храниться в различных форматах, однако наиболее распространенным с точки зрения использования в системах анализа данных является формат Comma Separated Values (CSV). Это простой текстовый формат, в котором данные разделены с помощью определенного символа (как правило, «,» или «;» – в зависимости от региональных настроек операционной системы). Для загрузки и манипуляции с табличными данными, как правило, используется пакет Pandas. Фрагмент кода для подключения необходимых библиотек и загрузке наборов данных для обучения и тестирования приведен на рис. 6.

```
In [1]: # Подключение необходимых библиотек для анализа данных
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Метод опорных векторов
from sklearn.svm import SVR

# Полиномиальные модели
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline

def PolynomialRegression(degree=2, **kwargs):
    return make_pipeline(PolynomialFeatures(degree),
                        LinearRegression(**kwargs))

In [2]: # Загрузка исходных данных
data_train = pd.read_csv("D:\\DataSet\\corrosion-control\\data-train.csv")
data_test = pd.read_csv("D:\\DataSet\\corrosion-control\\data-test.csv")
# Просмотр первых строк таблиц данных
```

Рис. 6. Подключение библиотек и загрузка исходных данных

Однако в исходном виде данные не могут быть использованы. Это объясняется тем, что первый столбец в таблицах данных содержит в себе текстовые метки, соответствующие наименованиям марок стали, которые не могут быть переданы в модель для обучения в первоначальном виде. Поэтому на первом этапе предварительной обработки данных каждой марки стали присваивается свой уникальный числовой индекс (рис. 7).

Далее размеченные данные разделяются на две части. Первая часть представляет собой наборы признаков (для рассматриваемой задачи набор признаков состоит из марки стали, солености морской воды и содержания в стали различных примесей). Эти данные будут использоваться в качестве входных наборов. Последние два столбца – потенциалы стали с оксидной пленкой и без нее, будут использованы в качестве отдельных выходных векторов и представляют собой сигналы на выходе вычислительных моделей. Таким образом, будут построены отдельные модели для вычисления потенциала с оксидной пленкой и без оксидной пленки (рис. 8). Для разбивки данных на входные и выходные выборки также наилучшим образом подходит библиотека Pandas.



```
In [3]: print("Данные для обучения")
data_train.head()

Данные для обучения

Out[3]:
   steelmark  salinity  carbon  manganese  chromium  silicon  nickel  copper  titanium
0         Ст3      18     0.18         0.53         0.30         0.23         0.30         0.3         0
1         09Г2      18     0.12         1.60         0.30         0.27         0.30         0.3         0

In [6]: print("Данные для обучения после разметки")
data_train.head()

Данные для обучения после разметки

Out[6]:
   steelmark  salinity  carbon  manganese  chromium  silicon  nickel  copper  titanium
0          1      18     0.18         0.53         0.30         0.23         0.30         0.3         0.0
1          2      18     0.12         1.60         0.30         0.27         0.30         0.3         0.0
2          3      18     0.21         0.80        13.00         0.80         0.00         0.0         0.0
3          4      18     0.12         0.65         0.75         0.95         0.65         0.5         0.0
4          5      18     0.12         2.00        18.00         0.80        10.00         0.0         0.7
```

Рис. 7. Данные для обучения модели до и после разметки

```
In [8]: ## Разбивка данных на входные и выходные последовательности
# Марка стали, соленость, содержание стали
X = data_train.iloc[:, 0:9]
# Потенциал стали с оксидной пленкой
Y1 = data_train.iloc[:, 9]
# Потенциал стали без оксидной пленкой
Y2 = data_train.iloc[:, 10]
X_test = data_test.iloc[:, 0:9]
Y1_test = data_test.iloc[:, 9]
Y2_test = data_test.iloc[:, 10]
```

Рис. 8. Разбивка данных на входные и выходные последовательности

Построение вычислительных моделей будет выполнено с помощью функций библиотеки Scikit-Learn, пожалуй, наиболее мощной библиотеки для машинного обучения в Python. Первая группа моделей будет построена на базе полиномиальной регрессии с различной степенью аппроксимирующих полиномов.

Были построены модели 3-й, 5-й и 7-й степени (рис. 9).

```
In [11]: ## Построение полиномиальных моделей
# Построение модели 2-й степени
Y1_predict_2 = PolynomialRegression(2).fit(X, Y1).predict(X_test)
Y2_predict_2 = PolynomialRegression(2).fit(X, Y2).predict(X_test)
# Построение модели 3-й степени
Y1_predict_3 = PolynomialRegression(3).fit(X, Y1).predict(X_test)
Y2_predict_3 = PolynomialRegression(3).fit(X, Y2).predict(X_test)
# Построение модели 5-й степени
Y1_predict_5 = PolynomialRegression(5).fit(X, Y1).predict(X_test)
Y2_predict_5 = PolynomialRegression(5).fit(X, Y2).predict(X_test)
```

Рис. 9. Построение моделей на основе полиномиальной регрессии 3-го, 5-го и 7-го порядков

Для оценивания точности полученных результатов была использована стандартная мера оценки – оценка относительной погрешности (рис. 10, табл. 3)

Таблица 3

**Относительная ошибка вычислений для разных марок стали и различных степеней регрессионного полинома**

		С оксидной пленкой					
		2-я степень		3-я степень		5-я степень	
Марка	Потенциал	Потенциал	Ошибка	Потенциал	Ошибка	Потенциал	Ошибка

стали	истинный	расчетный		расчетный		расчетный	
12X18H10T	362	439.16	21.3 %	444.95	22.9 %	407.13	12.5 %
Ст3	538	516.09	4.1 %	536.45	0.3 %	564.25	4.9 %
09Г2	502	527.45	5.1 %	527.34	5.1 %	518.94	3.4 %
20Х13	167	157.86	5.5 %	144.38	13.5 %	140.16	16.1 %
10ХСНД	533.67	536.53	0.5 %	531.81	0.3 %	586.27	9.9 %
Без оксидной пленки							
		2-я степень		3-я степень		5-я степень	
Марка стали	Потенциал истинный	Потенциал расчетный	Ошибка	Потенциал расчетный	Ошибка	Потенциал расчетный	Ошибка
12X18H10T	520	498.96	4.1 %	508.98	2.1 %	462.01	11.2 %
Ст3	434.33	434.62	0.1 %	434.38	0.1 %	441.69	1.7 %
09Г2	428.33	447.18	4.4 %	441.17	2.9 %	439.96	2.7 %
20Х13	459.00	494.97	7.8 %	506.50	10.3 %	494.11	7.7 %
10ХСНД	446.67	447.55	0.2 %	439.58	16 %	458.44	2.6 %

Как показывает анализ полученных результатов, оптимальным представляется использование различных вычислительных моделей с различными степенями регрессионных полиномов. Так, например, для марки стали 12X18H10T наилучшая точность получена с помощью полинома 5-й степени, а для марки Ст3 – с помощью полинома 3-й степени (рис. 11, 12).

```
In [12]: np.concatenate([Y1_predict_2, Y1_predict_3, Y1_predict_5])
Out[12]: array([439.15625, 516.0859375, 527.4453125, 157.85546875,
536.53125, 444.95467213, 536.45467213, 527.33571429,
144.38032787, 531.81032787, 407.13245402, 564.24787515,
518.9355001, 140.15914239, 586.2727591 ])

In [13]: ## Расчет ошибки предсказания модели
# С оксидной пленкой
E_Y1_2 = 100*np.abs((Y1_predict_2 - Y1_test.values) / Y1_test.values)
E_Y1_3 = 100*np.abs((Y1_predict_3 - Y1_test.values) / Y1_test.values)
E_Y1_5 = 100*np.abs((Y1_predict_5 - Y1_test.values) / Y1_test.values)
# Без оксидной пленки
E_Y2_2 = 100*np.abs((Y2_predict_2 - Y2_test.values) / Y2_test.values)
E_Y2_3 = 100*np.abs((Y2_predict_3 - Y2_test.values) / Y2_test.values)
E_Y2_5 = 100*np.abs((Y2_predict_5 - Y2_test.values) / Y2_test.values)
```

Рис. 10. Результаты расчетов и вычисление относительной ошибки

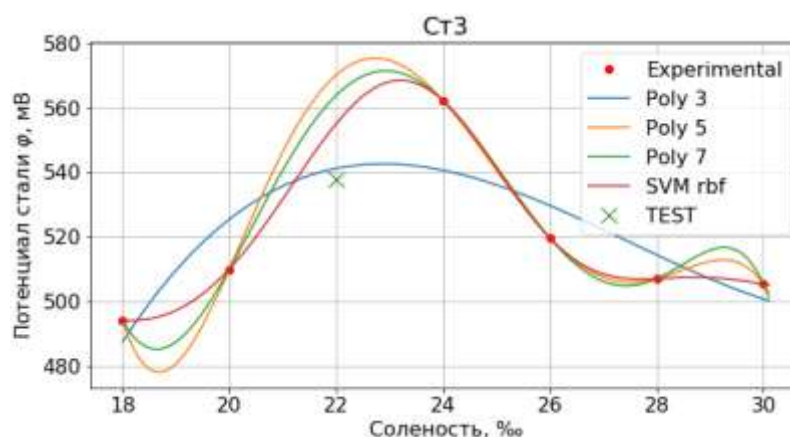


Рис. 11. Результат работы вычислительной модели с различными степенями регрессионных полиномов при прогнозировании потенциала Ст3 с оксидной пленки

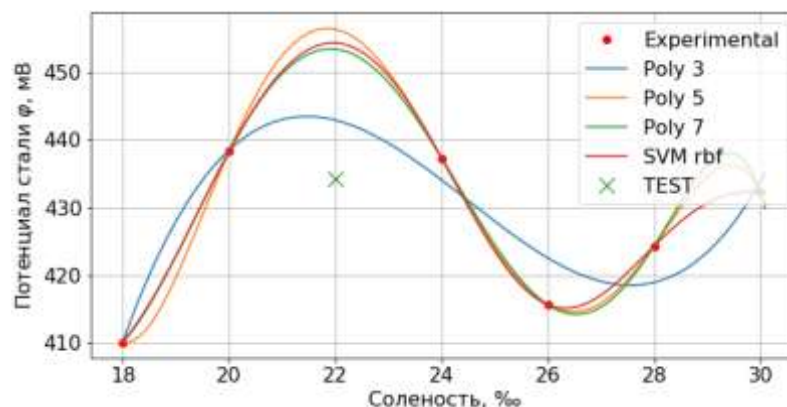


Рис. 12. Результат работы вычислительной модели с различными степенями регрессионных полиномов при прогнозировании потенциала Ст3 без оксидной пленки

Второй подход к построению вычислительных моделей основывался на использовании метода опорных векторов SVM (рис. 13).

```
In [16]: ## Решение с помощью вектора опорных векторов
svr_rbf = SVR(kernel='rbf', C=1000, gamma=0.1, epsilon=.01)
svr_poly = SVR(kernel='poly', C=100, gamma='auto', degree=3, epsilon=.0)
# Построение модели 2-й степени
Y1_predict_svr_rbf = svr_rbf.fit(X, Y1).predict(X_test)
Y2_predict_svr_rbf = svr_rbf.fit(X, Y2).predict(X_test)
Y1_predict_svr_poly = svr_poly.fit(X, Y1).predict(X_test)
Y2_predict_svr_poly = svr_poly.fit(X, Y2).predict(X_test)
# Вычисление ошибки модели
E_Y1_svr_rbf = 100*np.abs((Y1_predict_svr_rbf - Y1_test.values) / Y1_te
E_Y2_svr_rbf = 100*np.abs((Y2_predict_svr_rbf - Y2_test.values) / Y2_te
E_Y1_svr_poly = 100*np.abs((Y1_predict_svr_poly - Y1_test.values) / Y1
E_Y2_svr_poly = 100*np.abs((Y2_predict_svr_poly - Y2_test.values) / Y2
```

Рис. 13. Построение вычислительной модели на основе метода опорных векторов

При построении вычислительной модели использовались следующие разновидности – с полиномиальной ядерной функцией и ядром в виде радиальной базисной функции. Полученные результаты расчетов представлены в Таблице 4.

Таблица 4

**Относительная ошибка вычислений для разных марок стали и различных ядерных функций для метода опорных векторов**

С оксидной пленкой					
		polynomial		rbf	
Марка стали	Потенциал истинный	Потенциал расчетный	Ошибка	Потенциал расчетный	Ошибка
12X18H10T	362	438.81	21.2 %	449.37	24.1 %
Ст3	538	518.96	3.5 %	542.56	0.9 %
09Г2	502	540.39	7.6 %	525.85	4.8 %
20X13	167	132.53	20.6 %	151.17	9.5 %
10ХСНД	533.67	557.86	4.5 %	565.52	5.9 %
Без оксидной пленки					
		polynomial		rbf	
Марка стали	Потенциал истинный	Потенциал расчетный	Ошибка	Потенциал расчетный	Ошибка
12X18H10T	520	505.15	2.8 %	470.65	9.5 %
Ст3	434.33	431.13	0.7 %	458.97	5.7 %
09Г2	428.33	446.18	4.2 %	436.17	1.8 %
20X13	459.00	522.0	13.7 %	528.59	15.2 %

10XCHД	446.67	457.47	2.4 %	457.35	2.4 %
--------	--------	--------	-------	--------	-------

По аналогии с полиномиальным методом, результаты с наименьшей точностью соответствуют маркам стали 12X18H10T и 20X13 с оксидной пленкой, в тоже время точность оценки улучшилась для стали марки 09Г2 и Ст3.

### Заключение

Построена вычислительная модель значений потенциалов различных видов корпусных сталей с оксидной пленкой и без оксидной пленки при различной солёности морской воды, применяемых для корпусов морских судов и подводных конструкций океанотехнических сооружений, проектируемых для Черноморского бассейна Севастопольского региона и эксплуатирующихся в этом регионе.

Анализ результатов проведенного исследования показал, что оптимальным представляется использование различных вычислительных моделей с различными степенями регрессионных полиномов для достижения наилучшей точности для различных марок корпусных сталей.

### Благодарности

Исследование выполнено при финансовой поддержке РФФИ и города Севастополь в рамках научного проекта № 18-48-920017.

### Список литературы:

1. Kramar Vadim, Neural Network Modeling the Potential of Steels with Different Chrome Content / Vadim Kramar, Veronika Dushko, Anna Rodkina, Vasily Alchakov // Proceedings Engineering of the 26th International DAAAM Symposium «Intelligent Manufacturing & Automation» 21-24th October 2015, Zadar, Croatia, EU, Volume 26, No.1, ISSN 1726-9679, Ed. B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2016. – pp. 1059–1067.
2. Hussein Kadhim Mohammed AL-Shareefi. Neural Network Corrosion Control by Impressed Cathodic Protection / University of Technology – Baghdad – Iraq. 2009.
3. Wasserman, P.D. «Neural Computing :Theory and Practice», Van Nostrand Reinhold, New York, USA, 1989.
4. Jain, A.K., Mao, J.and Mohiuddin, K. «Artificial Neural Networks: A Tutorial», IEEE Computer, Vol.29, No.3, pp.31-44, 1996.
5. Hornik, K., Stinchcombe, M.and White, H. «Multilayer Feedforward Networks are Universal Approximators», Neural Networks, Vol.2, pp.359 -366, 1989.
6. Principe, J.C., Euliano, N.R.and Lefebvre, W.C. «Neural and Adaptive Systems», John Wiley and Sons, Inc., USA, 2000.
7. Hecht-Nielsen, R. «Neurocomputing», Addison-Wesley Publishing Company, Inc., 1990.
8. Shayma Mehdi Salih, «study the effect of some variables on cathodic protection current density» MSc.Thesis, Iraq University of Technology, 2005.
9. V A Kramar, V V Alchakov, V R Dushko, T V Kramar Application of support vector machine for prediction and classification / IOP Conf. Series: Journal of Physics: Conf. Series 1015 (2018) 032070 doi :10.1088/1742-6596/1015/3/032070
10. Круглов В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов – М.: Горячая линия – Телеком, 2002. – 382 с.
11. Вандер Плас Дж. Python для сложных задач: наука о данных и машинное обучение / Плас Дж. Вандер. – СПб.: Питер, 2018. – 576 с.: ил.
12. Родькина А.В. Гибридная нейронная сеть для прогнозирования защитного потенциала подводной части океанотехнических сооружений / А.В. Родькина, В.А. Крамарь, Е.М. Грамузов, О.А. Иванова // Труды Крыловского государственного научного центра. – СПб.: ФГУП «Крыловский государственный научный центр», 2019. – Журнал\_№ 20192. – Том 2. – С. 254–262.

## COMPUTATIONAL MODEL OF HULL STEEL POTENTIAL VALUES IN SEA WATER

**Vasily V. Alchakov**

*Sevastopol State University, Sevastopol, Russia*

**Vadim A. Kramar**

*Sevastopol State University, Sevastopol, Russia*

**Anna V. Rodkina**

*Sevastopol State University, Sevastopol, Russia*

*Abstract. The article provides an analysis of existing neural network models. The features of constructing a neural network using several parameters that affect the output value are described. The advantages of using neural networks and computing systems based on them are revealed. The task was solved using the Python programming language. The computing model of the potential values of steel with an oxide film and the potential of steel without an oxide film has been developed for various salinity of sea water and different types of hull steels used for hulls of sea vessels and underwater structures of ocean engineering structures designed for the Black Sea basin of the Sevastopol region and operated in this region. The obtained results will improve the accuracy of predicting potentials for various hull steel grades.*

Keywords: *neural networks, hull steels, sea vessels, ocean engineering structures, potential, layers, weight, thresholds.*

#### References:

1. Kramar Vadim, Dushko Veronika, Rodkina Anna, Alchakov Vasilii «Neural Network Modeling the Potential of Steels with Different Chrome Content» Proceedings Engineering of the 26th International DAAAM Symposium «Intelligent Manufacturing & Automation» 21-24th October 2015, Zadar, Croatia, EU, Volume 26, No.1, ISSN 1726-9679, Ed. B. Katalinic, Published by DAAAM International, Vienna, Austria, EU (2016): 1059–1067.
2. Hussein Kadhim Mohammed AL-Shareefi «Neural Network Corrosion Control by Impressed Cathodic Protection» University of Technology, Baghdad, Iraq (2009).
3. Wasserman P.D. «Neural Computing: Theory and Practice» Van Nostrand Reinhold, New York, USA (1989).
4. Jain A.K., Mao J., Mohiuddin K. «Artificial Neural Networks:A Tutorial» IEEE Computer, Vol. 29, No.3 (1996): 31–44.
5. Hornik K., Stinchcombe M., White H. «Multilayer Feedforward Networks are Universal Approximators» Neural Networks, Vol. 2 (1989): 359–366.
6. Principe J.C., Euliano N.R., Lefebvre W.C. «Neural and Adaptive Systems» John Wiley and Sons, Inc., USA (2000).
7. Hecht-Nielsen R.»Neurocomputing» Addison-Wesley Publishing Company, Inc. (1990).
8. Shayma Mehdi Salih, «Study the effect of some variables on cathodic protection current density» MSc.Thesis, Iraq University of Technology (2005).
9. Kramar V.A., Alchakov V.V., Dushko V.R., Kramar T.V. «Application of support vector machine for prediction and classification» IOP Conf. Series: Journal of Physics: Conf. Series 1015 (2018) 032070 doi :10.1088/1742-6596/1015/3/032070
10. Kruglov V.V., Borisov V.V. «Artificial neural networks. Theory and practice» M.: Hot line, Telecom (2002): 382.
11. Vander Place J. «Python for complex tasks: data science and machine learning» SPb.: Peter (2018): 576.
12. Rodkina A.V., Kramar V.A., Gramuzov E.M., Ivanova O.A. «Hybrid neural network for predicting the protective potential of the underwater part of ocean engineering structures» Proceedings of the Krylov State Scientific Center. SPb.: Federal State Unitary Enterprise «Krylov State Scientific Center», Journal\_№ 20192, Volume 2 (2019): 254–262.

#### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Альчаков Василий Викторович**, канд. техн. наук, доцент кафедры «Информатика и управление в технических системах» ФГАОУ ВО «Севастопольский государственный университет»,  
299053, г. Севастополь, ул. Университетская, 33  
e-mail: alchakov@sevsu.ru

**Крамарь Вадим Александрович**, д-р техн. наук, профессор кафедры «Информатика и управление в технических системах» ФГАОУ ВО «Севастопольский государственный университет»,  
299053, г. Севастополь, ул. Университетская, 33  
e-mail: vakramar@sevsu.ru

**Родкина Анна Владимировна**, канд. техн. наук, старший преподаватель кафедры «Океанотехника и кораблестроение» ФГАОУ ВО «Севастопольский государственный университет»,  
299053, г. Севастополь, ул. Университетская, 33  
e-mail: a.v.rodkina@gmail.com

**Vasilii V. Alchakov**, Ph.D. in Engineering Science, [Associate Professor](#) of the Department of Informatics and Control in Technical Systems, Sevastopol State University,  
33, Universitetskaya st., Sevastopol, 299053

**Vadim A. Kramar**, D.Sc.(Tech.), Professor of the Department of Informatics and Control in Technical Systems, Sevastopol State University,  
33, Universitetskaya st., Sevastopol, 299053

**Anna V. Rodkina**, Ph.D. in Engineering Science, Senior Lecturer of the Department of Ocean Technology and Shipbuilding Sevastopol State University,  
33, Universitetskaya st., Sevastopol, 299053